We claim:

Sub B1

1. On a computer having a memory, a method for debugging a program having a thread of execution, the method comprising: loading a debugger into a thread of execution of the program; and running the debugger in the thread of execution to debug the program.

2. The method of claim 1, wherein the program comprises at least one object, the method further comprises calling an interface of the object via the debugger.

3. The method of claim 2, wherein the program is executing on a first computer and the object is located on a second computer that is in communication with the first computer, the method further comprising calling a proxy interface via the debugger, wherein the proxy interface is located on the first computer and has a pointer to the object.

4. The method of claim 2, further comprising: creating a socket for communicating with the debugger; and sending commands through the socket to the debugger for conversion into function calls to the object interface.

5. The method of claim 2, wherein the object is a COM object.

6. The method of claim 2, wherein the object is a DCOM object.

Sub a1

7. The method of claim 1, wherein the program executes within a process defined within the memory, the method further comprising: establishing communication with a console module located outside of the process; receiving a command from the console; and converting the command into functions calls to the object interface.

8. A computer-readable medium having stored thereon computer-executable instructions for performing the method of claim 1.

9. A computer-readable medium having stored thereon computer-executable instructions for performing the method of claim 2.

10. A computer-readable medium having stored thereon computer-executable instructions for performing the method of claim 3.

11. A computer-readable medium having stored thereon computer-executable instructions for performing the method of claim 4.

12. A computer-readable medium having stored thereon computer-executable instructions for performing the method of claim 5.

13. A computer-readable medium having stored thereon computer-executable instructions for performing the method of claim 6.

14. A computer-readable medium having stored thereon computer-executable instructions for performing the method of claim 7.

15. On a computer having a memory, a method for debugging a program, the program having a thread of execution within the memory, the thread being associated with context data for describing the context of the thread, the method comprising: halting the thread of execution; obtaining a pointer to an interface of an object of the program from the context data; and referencing the pointer to make function calls manually to the interface from within the context of the thread.

16. The method of claim 15, wherein the program executes within a process defined within the memory, the method further comprising: establishing communication with a console module located outside of the process; receiving a command from the console; and converting the command into functions calls to the object interface.

17. The method of claim 16, wherein the establishing step comprises: creating a socket within the context of the thread; and communicating with the console module via the socket.

Sub a1

18. The method of claim 15, wherein the object is a COM object.

19. The method of claim 15, wherein the object is a DCOM object.

20. A computer-readable medium having stored thereon computer-executable instructions for performing the method of claim 15.

21. A computer-readable medium having stored thereon computer-executable instructions for performing the method of claim 16.

22. A computer-readable medium having stored thereon computer-executable instructions for performing the method of claim 17.

23. On a computer having a memory, a system for communicating with an object that is accessible by a program having a thread of execution within the memory, the thread being associated with context data for describing the context of the thread, the system comprising: a debugger module operating within the context of the thread; a socket accessible by the debugger module for sending and receiving messages; and a console module operating outside of the context of the thread for receiving a command from a user and sending the command to the debugger via the socket, wherein the debugger converts the command into a function call to the object.

Sub a1

24. The system of claim 23, wherein the debugger module is one of a plurality of debugger modules, the system further comprising: a multiplexor module for multiplexing commands from the console module to each of the plurality of debugger modules.

25. The system of claim 24, wherein each of the plurality of debugger modules is associated with a socket session, and the commands entered at the console module are multiplexed to the appropriate debugger module based on the socket session.

26. A method for debugging a mission-critical program, the method comprising: accessing the computer on which the program is running via a public network; halting a thread of execution of the program; allowing other threads of execution to continue; loading a debugger into the program's thread of execution; and running the debugger in the program's thread of execution to debug the program.

27. The method of claim 26, wherein the mission-critical program is executing on a web server.

28. The method of claim 26, wherein the mission-critical server is an electronic commerce program.

Sub a1

29. A computer-readable medium having stored thereon computer-executable instructions for performing the method of claim 26.

30. A computer-readable medium having stored thereon computer-executable instructions for performing the method of claim 27.

31. A computer-readable medium having stored thereon computer-executable instructions for performing the method of claim 28.

add a1